Tractability Frontier of Data Complexity in Team Semantics

Durand, Arnaud

2022-01

Durand , A , Kontinen , J , de Rugy-Altherre , N & Väänänen , J 2022 , ' Tractability Frontier of Data Complexity in Team Semantics ' , ACM Transactions on Computational Logic , vol. 23 , no. 1 , 3 , pp. 1-21 . https://doi.org/10.1145/3471618

http://hdl.handle.net/10138/338063
https://doi.org/10.1145/3471618

unspecified
acceptedVersion

Tractability Frontier of Data Complexity in Team Semantics

ARNAUD DURAND, IMJ-PRG, CNRS UMR 7586, Université de Paris, France
JUHA KONTINEN, Department of Mathematics and Statistics, University of Helsinki, Finland
NICOLAS DE RUGY-ALTHERRE, Université de Lorraine, France
JOUKO VÄÄNÄNEN, Department of Mathematics and Statistics, University of Helsinki, Finland and Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands

We study the data complexity of model-checking for logics with team semantics. We focus on dependence, inclusion, and independence logic formulas under both strict and lax team semantics. Our results delineate a clear tractability/intractability frontiers in data complexity of both quantifier-free and quantified formulas for each of the logics. For inclusion logic under the lax semantics, we reduce the model-checking problem to the satisfiability problem of so-called *dual-Horn* Boolean formulas. Via this reduction, we give an alternative proof for the known result that the data complexity of inclusion logic is in PTIME.

ACM Reference Format:
Arnaud Durand, Juha Kontinen, Nicolas de Rugy-Altherre, and Jouko Väänänen . 2021. Tractability Frontier of Data Complexity in Team Semantics. *ACM Trans. Comput. Logic* 1, 1 (June 2021), 21 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In this article we study the data complexity of model-checking of dependence, independence, and inclusion logic formulas. Independence and inclusion logic [3, 10] are variants of dependence logic [25] that extends first-order logic by dependence atoms of the form $=(x_1, \ldots, x_n)$ expressing that the value of x_n is functionally determined by the values of the variables $x_1, \ldots, x_{n-1}$. In independence and inclusion logic dependence atoms are replaced by independence and inclusion atoms $\overline{y} \perp_{\overline{x}} \overline{z}$ and $\overline{x} \subseteq \overline{y}$, respectively. The meaning of the independence atom is that, with respect to any fixed value of $\overline{x}$, the variables $\overline{y}$ are independent of the variables $\overline{z}$, whereas the inclusion atom expresses that all the values of $\overline{x}$ appear also as values for $\overline{y}$.

Team semantics is a framework for formalizing and studying various notions of dependence and independence pervasive in many areas of science. Team semantics differs from Tarski's semantics by interpreting formulas using sets of assignments instead of single assignments as in first-order logic. Reflecting this, dependence logic has higher expressive power than classical logics used for these purposes previously. Dependence, inclusion, and independence atoms are intimately connected to the corresponding functional, inclusion, and multivalued dependencies studied in database theory, see, e.g., [14]. Interestingly, independence atoms correspond naturally to a qualitative analogue of the

Authors' addresses: Arnaud Durand IMJ-PRG, CNRS UMR 7586, Université de Paris, France, durand@math.univ-paris-diderot.fr; Juha Kontinen Department of Mathematics and Statistics, University of Helsinki, Finland; Nicolas de Rugy-Altherre Université de Lorraine, France; Jouko Väänänen Department of Mathematics and Statistics, University of Helsinki, FinlandInstitute for Logic, Language and Computation, University of Amsterdam, The Netherlands.

notion of conditional independence in statistics [7]. On the other hand, team semantics can be naturally generalized to a probabilistic variant in which probabilistic independence can be taken as an atomic formula (see [1, 12] for further details).

Dependence logic and its variants can be used to formalize and study dependence and independence notions in various areas. For example, in the foundations of quantum mechanics, there are a range of notions of independence playing a central role in celebrated No-Go results such as Bell's theorem. Similarly, in the foundations of social choice theory, there are results such as Arrow's Theorem which can also be formalized in the team semantics setting [16, 22].

For the applications it is important to understand the complexity theoretic aspects of dependence logic and its variants. During the past few years, these aspects have been addressed in several studies. We will next briefly discuss some previous work. The data complexity of inclusion logic is sensitive to the choice between the two main variants of team semantics: under the so-called lax semantics it is equivalent to positive greatest fixed point logic (GFP^+) and captures PTIME over finite (ordered) structures [5]. Recently a fragment of inclusion logic that captures NL has been identified in [11]. The same article also exhibits surprisingly simple formulas of inclusion logic whose data complexity is NL and P-complete (see equations (2) and (3)). On the other hand, under the strict semantics, inclusion logic is equivalent to ESO (existential second order logic) and hence captures NP [4].

In [2] the fragment of dependence logic allowing only sentences in which dependence atoms of arity at most k may appear (atoms $=(x_1, \ldots, x_n)$ satisfying $n \leq k+1$) was shown to correspond to the k-ary fragment $\mathrm{ESO}_f(k\text{-ary})$ of ESO in which second-order quantification is restricted to at most k-ary functions and relations. Several similar results have been obtained also for independence and inclusion logic, e.g., in [4, 13, 23, 24].

The combined complexity of the model-checking problem of dependence logic, and many of its variants, was shown to be NEXPTIME-complete [9]. On the other hand, the satisfiability problem for the two variable fragment of dependence logic (and many of its variants) was shown to be NEXPTIME-complete in [19, 20]. Furthermore, during the past few years, the complexity aspects of propositional and modal logics in team semantics have been also systematically studied (see [15, 21] and the references therein).

The starting point for the present work are the following results of [18] showing that the non-classical interpretation of disjunction in team semantics makes the model-checking of certain quantifier-free formulas very complicated. Define ϕ_1 and ϕ_2 as follows:

(1) ϕ_1 is the formula $=(x, y) \vee =(u, v)$, and
(2) ϕ_2 is the formula $=(x, y) \vee =(u, v) \vee =(u, v)$.

Surprisingly, the data complexity of the model-checking problem of ϕ_1 and ϕ_2 are already NL-complete and NP-complete, respectively. In [18] it was also shown that model-checking for $\phi \vee \psi$ where ϕ and ψ are 2-*coherent* quantifier-free formulas of dependence logic is always in NL. A formula ϕ is called k-coherent if, for all $\mathfrak{A}$ and X, $\mathfrak{A} \models_X \phi$, if and only if, $\mathfrak{A} \models_Y \phi$ for all $Y \subseteq X$ such that $|Y| = k$. Note that the left-to-right implication is always true due to the downwards closure property of dependence logic formulas. The downwards closure property also implies that, for dependence logic formulas, the strict and the lax semantics are equivalent. For independence and inclusion logic formulas this is not the case.

In this article our goal is to shed light on the tractability frontier of data complexity of dependence, independence, and inclusion logic formulas under both strict and lax team semantics. In order to state our results, we define a new syntactic measure called the disjunction-width $\mathrm{d}_\vee(\phi)$ of a formula ϕ. Our results show that, for quantifier-free formulas ϕ of dependence logic, the data complexity of model-checking is in NL if $\mathrm{d}_\vee(\phi) \leq 2$. Surprisingly, for independence

logic the case of quantifier-free formulas turns out to be more fine-grained. In particular, we exhibit a quantifier-free formula with $d_\vee(\phi) \leq 2$ whose data-complexity is NP-complete and also identify a more restricted fragment with data complexity in NL. For quantified formulas, the complexity is shown to be NP-complete already with simple formulas constructed in terms of existential quantification and conjunction in the empty non-logical vocabulary.

For inclusion logic, we show that model-checking can be reduced to the satisfiability problem of dual-Horn propositional formulas. While interesting in its own right, this also provides an alternative proof for the fact (see [5]) that the data complexity of inclusion logic is in PTIME, and is also analogous to the classical result of Grädel on the Horn fragment of second-order logic [8]. We also show that, under the strict semantics, the tractability frontier of model-checking of (both quantifier-free and quantified) inclusion logic formulas becomes similar to that of dependence and independence logic.

2 PRELIMINARIES

In this section we briefly discuss the basic definitions and results needed in this article.

Definition 1. Let $\mathfrak{A}$ be a structure with domain A, and $V = \{x_1, \ldots, x_k\}$ be a finite (possibly empty) set of variables.

- A *team* X of $\mathfrak{A}$ with domain $\mathrm{Dom}(X) = V$ is a finite set of assignments $s \colon V \to A$.
- For a tuple $\overline{x} = (x_1, \ldots, x_n)$, where $x_i \in V$, $X(\overline{x}) := \{s(\overline{x}) : s \in X\}$ is the n-ary relation of A, where $s(\overline{x}) := (s(x_1), \ldots, s(x_n))$.
- For $W \subseteq V$, $X \upharpoonright W$ denotes the team obtained by restricting all assignments of X to W.
- The set of free variables of a formula ϕ is defined as in first-order logic, taking into account that free variables may arise also from dependence, independence and inclusion atoms, and is denoted by $\mathrm{Fr}(\phi)$.

We will consider two variants of the semantics called the strict and the original semantics given in [25] is a combination of these variants (with the lax disjunction and the strict existential quantifier). For dependence logic formulas, the two variants of the semantics are easily seen to be equivalent, but for independence and inclusion logic this is not the case. We first define the lax team semantics for first-order formulas in negation normal form. Below $\mathfrak{A} \models_s \alpha$ refers to the satisfaction in first-order logic, and $s(m/x)$ is the assignment such that $s(m/x)(x) = m$, and $s(m/x)(y) = s(y)$ for $y \neq x$. The power set of a set A is denoted by $\mathcal{P}(A)$.

Definition 2. Let $\mathfrak{A}$ be a structure, X be a team of A, and ϕ be a first-order formula such that $\mathrm{Fr}(\phi) \subseteq \mathrm{Dom}(X)$.

lit: For a first-order literal α, $\mathfrak{A} \models_X \alpha$ if and only if, for all $s \in X$, $\mathfrak{A} \models_s \alpha$.

$\vee$: $\mathfrak{A} \models_X \psi \vee \theta$ if and only if, there are Y and Z such that $Y \cup Z = X$, $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \theta$.

$\wedge$: $\mathfrak{A} \models_X \psi \wedge \theta$ if and only if, $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \theta$.

$\exists$: $\mathfrak{A} \models_X \exists x\psi$ if and only if, there exists a function $F : X \to \mathcal{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X(F/x)} \psi$, where $X(F/x) = \{s(m/x) : s \in X, m \in F(s)\}$.

$\forall$: $\mathfrak{A} \models_X \forall x\psi$ if and only if, $\mathfrak{A} \models_{X(A/x)} \psi$, where $X(A/x) = \{s(m/x) : s \in X, m \in A\}$.

A sentence ϕ is *true* in $\mathfrak{A}$ (abbreviated $\mathfrak{A} \models \phi$) if $\mathfrak{A} \models_{\{\emptyset\}} \phi$. Sentences ϕ and ϕ' are *equivalent*, $\phi \equiv \phi'$, if for all models $\mathfrak{A}$, $\mathfrak{A} \models \phi \Leftrightarrow \mathfrak{A} \models \phi'$.

In the *Strict Semantics*, the semantic rule for disjunction is modified by adding the requirement $Y \cap Z = \emptyset$, and the clause for the existential quantifier is replaced by

> $\mathfrak{A} \models_X \exists x\psi$ if and only if, there exists a function $H : X \to A$ such that $\mathfrak{A} \models_{X(H/x)} \psi$, where $X(H/x) = \{s(H(s)/x) : s \in X\}$.

The meaning of first-order formulas is invariant under the choice between the strict and the lax semantics. First-order formulas satisfy what is known as the *flatness* property: $\mathfrak{A} \models_X \phi$, if and only if, $\mathfrak{A} \models_s \phi$ for all $s \in X$. Next we will give the semantic clauses for the new dependency atoms:

Definition 3.
- Let $\overline{x}$ be a tuple of variables and let y be another variable. Then $=(\overline{x}, y)$ is a *dependence atom*, with the semantic rule

  $\mathfrak{A} \models_X =(\overline{x}, y)$ if and only if for all $s, s' \in X$, if $s(\overline{x}) = s'(\overline{x})$, then $s(y) = s'(y)$;
- Let $\overline{x}$, $\overline{y}$, and $\overline{z}$ be tuples of variables (not necessarily of the same length). Then $\overline{x} \perp_{\overline{y}} \overline{z}$ is a *conditional independence atom*, with the semantic rule

  $\mathfrak{A} \models_X \overline{x} \perp_{\overline{y}} \overline{z}$ if and only if for all $s, s' \in X$ such that $s(\overline{y}) = s'(\overline{y})$, there exists an assignment $s'' \in X$ such that $s''(\overline{xyz}) = s(\overline{xy})s'(\overline{z})$.

  Furthermore, when $\overline{z}$ is empty, we write $\overline{x} \perp \overline{y}$ as a shorthand for $\overline{x} \perp_{\overline{z}} \overline{y}$, and call it a *pure independence atom*;
- Let $\overline{x}$ and $\overline{y}$ be two tuples of variables of the same length. Then $\overline{x} \subseteq \overline{y}$ is an *inclusion atom*, with the semantic rule

  $\mathfrak{A} \models_X \overline{x} \subseteq \overline{y}$ if and only if for all $s \in X$ there exists a $s' \in X$ such that $s'(\overline{y}) = s(\overline{x})$.

The formulas of dependence logic, D, are obtained by extending the syntax of FO by dependence atoms. The semantics of D-formulas is obtained by extending Definition 2 by the semantic rule defined above for dependence atoms. Independence logic, FO($\perp_c$), and inclusion logic, FO($\subseteq$), are defined analogously using independence and inclusion atoms, respectively.

It is easy to see that the flatness property is lost immediately when FO is extended by any of the atoms defined above. On the other hand, it is straightforward to check that all D-formulas satisfy the following strong *downwards closure* property: if $\mathfrak{A} \models_X \phi$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y \phi$. Another basic property of logics in team semantics is called *locality*:

Proposition 4 (Locality). *Let ϕ be a formula of any of the logics* D, FO($\perp_c$) *or* FO($\subseteq$). *Then, under the lax semantics, for all structures $\mathfrak{A}$ and teams X:*

$$\mathfrak{A} \models_X \phi \text{ iff } \mathfrak{A} \models_{X \restriction \mathrm{Fr}(\phi)} \phi.$$

Under the strict semantics locality holds only for dependence logic formulas (see [3] for details).

In this article we study the data complexity of model-checking of dependence, independence, and inclusion logic formulas. In other words, for a fixed formula ϕ of one of the aforementioned logics, we study the complexity of the following model-checking problem: given a finite structure $\mathfrak{A}$ and a team X, decide whether $\mathfrak{A} \models_X \phi$. Note that when we are working with the lax semantics, we may assume without loss of generality that the domain of X is exactly $\mathrm{Fr}(\phi)$. *If not explicitly mentioned otherwise, all results are valid under both strict and lax semantics.* We assume that the reader is familiar with the basics of computational complexity theory such as NP-completeness and the log-space bounded classes L and NL.

2.1 Complexity classes and satisfiability problems

In the paper, we will make use of some well-known computational problems whose complexity is recalled here (unless specified all proofs can be found in [6]). We suppose the reader familiar with basic complexity classes such as L (logarithmic space), NL (non deterministic logarithmic space), PTIME (polynomial time), NP (non deterministic polynomial time).

Let $k \in \mathbb{N}$, a propositional formula is in k-CNF if it is in conjunctive normal form with clauses of length at most k. It is positive, if it equivalent to a formula without negation.

It is well known that, k-SAT the satisfiability problem for k-CNF formulas is NP-complete for $k \geq 3$ and NL-complete for $k = 2$. HORN-SAT (resp. DUAL-HORN-SAT) is the satisfiability problem of CNF formulas with at most one positive (resp. negative) literal per clause. This problem is known to be PTIME-problem.

Given a positive 3-CNF formula, deciding whether ϕ is satisfiable such that exactly one variable is set to true in each clause is also known to be NP-complete. This problem is called 1-IN-3-SAT.

3 DEPENDENCE AND INDEPENDENCE LOGICS

In this section we consider the complexity of model-checking for quantifier-free and quantified formulas of dependence and independence logic.

3.1 The case of quantifier-free formulas

In this section we consider the complexity of model-checking for quantifier-free formulas of dependence and independence logic. For dependence logic the problem has already been essentially settled in [18]. The following theorems delineate a clear barrier between tractability and intractability for quantifier-free dependence logic formulas.

Theorem 5 ([18])**.** *The model checking problem for the formula*

$$=(x, y) \vee =(z, v)$$

is NL*-complete. More generally, the model-checking for $\phi \vee \psi$ where ϕ and ψ are 2-coherent quantifier-free formulas of* D *is always in* NL*.*

When two disjunctions can be used, the model checking problem becomes intractable as shown by the following results.

Theorem 6 ([18])**.** *The model checking problem for the formula*

$$=(x, y) \vee =(z, v) \vee =(z, v)$$

is NP*-complete.*

In order to give a syntactic generalization of Theorem 5, we define next the disjunction-width of a formula.

Definition 7. Let σ be a relational signature. The disjunction-width of a σ-formula ϕ, denoted $\mathrm{d}_\vee(\phi)$, is defined as follows:

$$\mathrm{d}_\vee(\phi) = \begin{cases} 1 \text{ if } \phi \text{ is } \overline{y} \perp_{\overline{x}} \overline{z} \text{ or } =(\overline{x}, y) \text{ or } \overline{x} \subseteq \overline{y} \\ 0 \text{ if } \phi \text{ is } R(\overline{x}) \text{ or } \neg R(\overline{x}), \text{ for } R \in \sigma \cup \{=\} \\ \max(\mathrm{d}_\vee(\phi_1), \mathrm{d}_\vee(\phi_2)) \text{ if } \phi \text{ is } \phi_1 \wedge \phi_2 \\ \mathrm{d}_\vee(\phi_1) + \mathrm{d}_\vee(\phi_2) \text{ if } \phi \text{ is } \phi_1 \vee \phi_2 \\ \mathrm{d}_\vee(\phi_1) \text{ if } \phi \text{ is } \exists x \phi_1 \text{ or } \forall x \phi_1. \end{cases}$$

The next theorem is a syntactically defined analogue of Theorem 5.

Proposition 8. *The data complexity of model-checking of quantifier-free* D*-formulas ϕ with $d_\vee(\phi) \leq 2$ is in* NL*.*

Proof. We will first show that a formula ϕ with $\mathrm{d}_\vee(\phi) = 1$ is 2-coherent. This follows by induction using the following facts [18]:

- dependence atoms are 2-coherent, and first-order formulas are 1-coherent,

- if ψ is k-coherent, then $\psi \vee \phi$ is also k-coherent assuming ϕ is first-order,
- if ψ is k-coherent and ϕ is $k \leq j$-coherent, then $\psi \wedge \phi$ is j-coherent.

It is also straightforward to check that the data complexity of a formula ϕ with $\mathrm{d}_\vee(\phi) = 1$ is in L (the formula ϕ can be expressed in FO assuming the team X with domain $\overline{x} = \mathrm{Fr}(\phi)$ is represented by the n-ary relation $X(\overline{x})$). We will complete the proof using induction on ϕ with $\mathrm{d}_\vee(\phi) = 2$. Suppose that $\phi = \psi \vee \eta$, where $\mathrm{d}_\vee(\psi) = \mathrm{d}_\vee(\eta) = 1$. Then the claim follows by Theorem 5. The case $\phi = \psi \wedge \eta$ is also clear. Suppose finally that $\phi = \psi \vee \eta$, where η is first-order. Note that by downward closure and flatness

$$\mathfrak{A} \models_X \phi \Leftrightarrow \mathfrak{A} \models_{X'} \psi,$$

where $X' = \{s \in X \mid \mathfrak{A} \not\models_s \eta\}$. Now since X' can be computed in L, the model-checking problem of ϕ can be decided in NL by the induction assumption for ψ. □

In the rest of this section we examine potential analogues of Theorems 5 and 6 for independence logic. It is well-known that the dependence atom $=(\overline{x}, y)$ is logically equivalent to the independence atom $y \perp_{\overline{x}} y$. Hence, the following is immediate from Theorem 6.

Corollary 9. *The model checking problem for the formula*

$$y \perp_x y \vee v \perp_z v \vee v \perp_z v$$

is NP*-complete.*

For independence logic, the situation is not as clear as for dependence logic concerning tractability. In the following we will exhibit a fragment of independence logic whose data complexity is in NL and which is in some sense the maximal such fragment.

Definition 10. The Boolean closure of an independence atom by first-order formulas, denoted $\mathrm{BC}(\perp, \mathrm{FO})$, is defined as follows:

- Any independence atom $\overline{x} \perp_{\overline{y}} \overline{z}$ is in $\mathrm{BC}(\perp, \mathrm{FO})$.
- If $\phi \in \mathrm{BC}(\perp, \mathrm{FO})$, then for any formula $\phi \in \mathrm{FO}$, $\phi \wedge \phi$ and $\phi \vee \phi$ are in $\mathrm{BC}(\perp, \mathrm{FO})$.

Let $\phi \in \mathrm{BC}(\perp, \mathrm{FO})$. Up to permutation of disjunction and conjunction, ϕ can be put into the following normalized form:

$$\phi \equiv ((\dots((\overline{x} \perp_{\overline{z}} \overline{y} \wedge \phi_1) \vee \psi_1) \wedge \dots) \wedge \phi_k) \vee \psi_k). \tag{1}$$

For a structure $\mathfrak{A}$, $\mathfrak{C}^+ := \bigcap_{i=1}^k \phi_i(\mathfrak{A})$ and $\mathfrak{C}^- := \bigcup_{i=1}^k \psi_i(\mathfrak{A})$, where $\phi_i(\mathfrak{A}) := \{s \mid \mathfrak{A} \models_s \phi_i\}$.

The next lemma gives a combinatorial criterion for the satisfaction of $\mathrm{BC}(\perp, \mathrm{FO})$ formulas.

Lemma 11. *For all $\phi \in \mathrm{BC}(\perp, \mathrm{FO})$ of the form (1), structures $\mathfrak{A}$, and teams X the following are equivalent:*

(A) $\mathfrak{A} \models_X \phi$.

(B) *(1) and (2) below hold:*

(1) *For all $s \in X$ either $s \in [((..(\psi_1 \wedge \phi_2) \vee \dots) \wedge \phi_k) \vee \psi_k](\mathfrak{A})$ or $s \in \mathfrak{C}^+ \setminus \mathfrak{C}^-$.*

(2) *For all $s_1, s_2 \in X$ such that $s_1, s_2 \in \mathfrak{C}^+ \setminus \mathfrak{C}^-$, and $s_1(\overline{z}) = s_2(\overline{z})$, there exists $s_3 \in X \cap \mathfrak{C}^+$, such that: $s_3(\overline{z}) = s_1(\overline{z}), s_3(\overline{x}) = s_1(\overline{x})$ and $s_3(\overline{y}) = s_2(\overline{y})$.*

Assignments s_1, s_2 as in (2) will be called compatible for formula ϕ and team X. Furthermore, s_3 as in (2) will be called a witness of s_1, s_2 for formula ϕ.

Proof. Note that (A) is equivalent to the existence of subteams X' and Y_i, $1 \leq i \leq k$, of X such that

(i)$_k$ $X = X' \cup Y_1 \cup \ldots \cup Y_k$,
(ii)$_k$ $\mathfrak{A} \models_{Y_i} \psi_i$, for $1 \leq i \leq k$,
(iii)$_k$ $\mathfrak{A} \models_{X' \cup \bigcup_{1 \leq j < i} Y_j} \phi_i$, for $1 \leq i \leq k$
(iv)$_k$ $\mathfrak{A} \models_{X'} \overline{x} \perp_{\overline{z}} \overline{y}$.

This can be proved by induction on k as follows: The claim is clearly true for $k = 1$. Suppose it holds for $k-1$, where $k > 1$. We know ϕ is equivalent to

$$(((\ldots((\overline{x} \perp_{\overline{z}} \overline{y} \wedge \phi_1) \vee \psi_1) \wedge \ldots) \wedge \phi_k) \vee \psi_k.$$

Hence $\mathfrak{A} \models_X \phi$ if and only if $X = Z \cup Y_k$ such that

$$\mathfrak{A} \models_Z (((\ldots((\overline{x} \perp_{\overline{z}} \overline{y} \wedge \phi_1) \vee \psi_1) \wedge \ldots) \wedge \phi_{k-1}) \vee \psi_{k-1}) \wedge \phi_k$$

and $\mathfrak{A} \models_{Y_k} \psi_k$. In particular, $\mathfrak{A} \models_Z \phi_k$ and

$$\mathfrak{A} \models_Z ((\ldots((\overline{x} \perp_{\overline{z}} \overline{y} \wedge \phi_1) \vee \psi_1) \wedge \ldots) \wedge \phi_{k-1}) \vee \psi_{k-1}.$$

By Induction Hypothesis there are subteams X' and Y_i, $1 \leq i < k$, of Z such that (i)$_{k-1}$-(iv)$_{k-1}$ hold with X replaced by Z. Now the sequence $X', Y_1, \ldots, Y_k$ satisfies (i)$_k$-(iv)$_k$ and is therefore as required for the induction claim. The converse is similar.

(A) implies (B): To prove (1), suppose $s \in X$. Thus $s \in X' \cup Y_1 \cup \ldots \cup Y_k$, where $X', Y_1, \ldots, Y_k$ are as defined earlier. Suppose

$$s \notin [((..(\psi_1 \wedge \phi_2) \vee \ldots) \wedge \phi_k) \vee \psi_k](\mathfrak{A}).$$

Then $\mathfrak{A} \not\models_s \psi_k$, whence $s \notin Y_k$. Hence $s \in X' \cup Y_1 \cup \ldots \cup Y_{k-1}$, whence $\mathfrak{A} \models_s \phi_k$, and therefore

$$s \notin [((..(\psi_1 \wedge \phi_2) \vee \ldots) \wedge \phi_{k-1}) \vee \psi_{k-1}](\mathfrak{A}).$$

Eventually, step by step, we verify $s \in \mathfrak{C}^+ \setminus \mathfrak{C}^-$ and hence (1) is proved.

To prove (2), suppose $s_1, s_2 \in X \cap (\mathfrak{C}^+ \setminus \mathfrak{C}^-)$ such that $s_1(\overline{z}) = s_2(\overline{z})$. By (i)$_k$-(iii)$_k$, $s_1, s_2 \in X'$. By (iv)$_k$ there is a witness as required.

(B) implies (A): Let us denote by $X' \subseteq (X \cap \mathfrak{C}^+)$ a minimal superset of $X \cap (\mathfrak{C}^+ \setminus \mathfrak{C}^-)$ satisfying $\overline{x} \perp_{\overline{z}} \overline{y}$. Such a set X' exists by the assumption (2). Then, for $1 \leq i \leq k$, define

$$Y_i = (X \setminus X') \cap [\psi_i \wedge \bigwedge_{i < j \leq k} \phi_j](\mathfrak{A}).$$

Suppose $s \in X$ but $s \notin (\mathfrak{C}^+ \setminus \mathfrak{C}^-) \subseteq X'$. By (1),

$$s \in [((..(\psi_1 \wedge \phi_2) \vee \ldots) \wedge \phi_k) \vee \psi_k](\mathfrak{A}).$$

If $s \notin Y_k$, then

$$s \in [((..(\psi_1 \wedge \phi_2) \vee \cdots \vee \psi_{k-1}) \wedge \phi_k)](\mathfrak{A}).$$

Hence $\mathfrak{A} \models_s \phi_k$. If $s \notin Y_{k-1}$, then

$$s \in [((..(\psi_1 \wedge \phi_2) \vee \cdots \vee \psi_{k-2}) \wedge \phi_{k-1})](\mathfrak{A}).$$

Hence $\mathfrak{A} \models_s \phi_{k-1}$. Continuing this way we see that if $s \notin Y_1 \cup \ldots Y_k$, then $s \in \mathfrak{C}^+ \setminus \mathfrak{C}^-$, contrary to the assumption $s \notin \mathfrak{C}^+ \setminus \mathfrak{C}^-$. We have proved $(\mathrm{i})_k$. Clearly $(\mathrm{ii})_k$ and $(\mathrm{iii})_k$ hold. Finally, $(\mathrm{iv})_k$ holds by the definition of X'. □

Theorem 12. *The data complexity of any $\phi \in \mathrm{BC}(\bot, \mathrm{FO})$ is in* L. *This is true both in lax and in strict semantics.*

Proof. It is well known that checking whether a tuple s belongs to the query result $\phi(\mathfrak{A})$ of a first-order formula can be done in logarithmic space [17]. Therefore, by the characterization of Lemma 11, deciding whether $\mathfrak{A} \models_X \phi$ is also in L. For the strict semantics, it suffices to note that the proof of Lemma 11 goes through also under the strict semantics. We can use the fact that the formulas ϕ_i have the downwards closure property to force the subteams $X', Y_1, .., Y_k$ in the decomposition of X to be pairwise disjoint. □

Interestingly the analogue of Lemma 11 does not hold for inclusion atoms; It was recently shown in [11] that the data complexity of the formula

$$x \subseteq y \vee u = v \tag{2}$$

is already NL-complete and for

$$(x \subseteq z \wedge y \subseteq z) \vee u = v \tag{3}$$

the problem becomes P-complete. These results hold under both strict and lax semantics as the other disjunct in both formulas is first-order and satisfies the downwards closure property. Furthermore, in the last section of this article we construct a quantifier free inclusion logic formula with NP-complete data complexity under the strict semantics.

Theorem 13. *The data complexity of formulae of the form $\phi_1 \vee \phi_2$ with $\phi_1, \phi_2 \in \mathrm{BC}(\bot, \mathrm{FO})$ is in* NL *under the lax semantics.*

Proof. The proof is given by a log-space reduction to the NL-complete problem 2-SAT. Given a structure $\mathfrak{A}$ and a team X we construct a 2-CNF propositional formula Φ such that:

$$\mathfrak{A} \models_X \phi_1 \vee \phi_2 \iff \Phi \text{ is satisfiable.} \tag{4}$$

Recall that if a team X is such that $\mathfrak{A} \models_X \phi_1 \vee \phi_2$ then, there exists $Y, Z \subseteq X$ such that $Y \cup Z = X$ and $\mathfrak{A} \models_Y \phi_1$ and $\mathfrak{A} \models_Z \phi_2$. For each assignment $s \in X$, we introduce two Boolean variables $Y[s]$ and $Z[s]$. Our Boolean formula Φ will be defined below with these $2|X|$ variables the set of which is denoted by $\mathrm{Var}(\Phi)$. It will express that the set of assignments must split into Y and Z but also make sure that incompatible (see Lemma 11) assignments do not appear in the same subteam.

For each pair s_i, s_j that are incompatible for ϕ_1 on team X, one adds the 2-clause: $\neg Y[s_i] \vee \neg Y[s_j]$. The conjunction of these clauses is denoted by C_Y. Similarly, for each pair s_i, s_j that are incompatible for ϕ_2 on team X, one adds the clause: $\neg Z[s_i] \vee \neg Z[s_j]$ and call C_Z the conjunction of these clauses.

Finally, the construction of ϕ is completed by adding the following conjunctions:

$$C^0 := \bigwedge \{Y[s] \vee Z[s] : s \in X\}$$
$$C^1 := \bigwedge \{\neg Y[s] : s \text{ fails (B,1) of Lemma 11 for } \phi_1\}$$
$$C^2 := \bigwedge \{\neg Z[s] : s \text{ fails (B,1) of Lemma 11 for } \phi_2\}$$

It is not hard to see that the formula

$$\Phi \equiv \bigwedge_{0 \leq i \leq 2} C^i \wedge C_Y \wedge C_Z$$

can be constructed deterministically in log-space. It remains to show that the equivalence (4) holds.

Assume that the left-hand side of the equivalence holds. Then, there exists $Y, Z \subseteq X$ such that $Y \cup Z = X$, $\mathfrak{A} \models_Y \phi_1$ and $\mathfrak{A} \models_Z \phi_2$. We construct a propositional assignment $I : \mathrm{Var}(\Phi) \to \{0, 1\}$ as follows. For all $s \in Y$, we set $I(Y[s]) = 1$ and for all $s \in Z$, we set similarly $I(Z[s]) = 1$. It is now immediate that the all of the clauses in $\bigwedge_{0 \leq i \leq 2} C^i$ are satisfied by I.

Let us consider a clause $\neg Y[s_i] \vee \neg Y[s_j]$ for an incompatible pair s_i, s_j. Then, $I(Y[s_i]) = 0$ or $I(Y[s_j]) = 0$ must hold. For a contradiction, suppose that $I(Y[s_i]) = I(Y[s_j]) = 1$. Then since $\mathfrak{A} \models_Y \phi_1$ holds, by construction s_i and s_j must be compatible for ϕ_1. Hence we get a contradiction and may conclude that I satisfies $\neg Y[s_i] \vee \neg Y[s_j]$. The situation is similar for each clause $\neg Z[s_i] \vee \neg Z[s_j]$.

Let us then assume that Φ is satisfiable, and let $I : \mathrm{Var}(\Phi) \to \{0, 1\}$ be a satisfying assignment for Φ. Since $I \models C^0$, we get that $I(Y[s]) = 1$ or $I(Z[s]) = 1$ for all $s \in X$. Let

$$X_Y = \{s : I(Y[s]) = 1\} \text{ and } X_Z = \{s : I(Z[s]) = 1\}.$$

Now $X_Y \cup X_Z = X$ and clauses C^1 (C^2) ensure that each $s \in X_Y$ ($s \in X_Z$) satisfies condition (B,1) of Lemma 11.

We will next show how the sets X_Y and X_Z can be extended to sets Y and Z satisfying also condition (B,2) of Lemma 11 and consequently $\mathfrak{A} \models_Y \phi_1$ and $\mathfrak{A} \models_Z \phi_2$. Note first that, since I satisfies Φ, for all $s_1, s_2 \in X_Y$, Φ cannot have a clause of the form $\neg Y[s_1] \vee \neg Y[s_2]$, and hence s_1, s_2 are compatible for ϕ_1. Analogously we see that all $s_1, s_2 \in X_Z$ are compatible for ϕ_2. We will define the sets Y and Z incrementally by first initializing them to X_Y and X_Z, respectively. Note that even if $X_Y \cup X_Z = X$, no decision has been made regarding the membership of assignments s in Y (resp. Z) such that $I(Y[s]) = 0$ (resp. $I(Z[s]) = 0$). Let us first consider Y. Until no changes occur, we consider all pairs $s_1, s_2 \in Y \cap (\mathfrak{C}^+ \setminus \mathfrak{C}^-)$ (where the sets $\mathfrak{C}^+$ and $\mathfrak{C}^-$ are taken with respect to ϕ_1) such that $s_1(\overline{z}) = s_2(\overline{z})$ and add into Y (if they are not already in) all tuples s_3 such that s_3 is a witness for the pair (s_1, s_2) regarding formula ϕ_1. Since by construction s_1, s_2 are compatible then at least one such s_3 exists (but may be initially outside of Y). We prove below that this strategy is safe. First of all, it is easily seen that any pair among $\{s_1, s_2, s_3\}$ is compatible for ϕ_1. Therefore, it remains to show that the new assignments s_3 are compatible with every other element s added to Y so far. Suppose this is not the case and that there exists $s \in (Y \cap (\mathfrak{C}^+ \setminus \mathfrak{C}^-)) \setminus \{s_1, s_2\}$ such that s_3 and s are incompatible for ϕ_1. Note that for incompatibility we must have $s_3(\overline{z}) = s(\overline{z})$. Since s_1, s_2, and s are in Y they are all pairwise compatible. Hence, there exists t_1 such that t_1 is a witness for the pair (s_1, s). Then, $t_1(\overline{x}) = s_1(x) = s_3(\overline{x})$, and $t_1(\overline{y}) = s(\overline{y})$. Consequently, t_1 is also a witness for s_3, s hence, s_3 and s are compatible which is a contradiction. Therefore, the assignment s_3 can be safely added to Y. The set Z is defined analogously. By the construction, the sets Y and Z satisfy conditions (B,1) and (B,2) of Lemma 11 for ϕ_1 and ϕ_2, respectively, and hence it holds that $\mathfrak{A} \models_Y \phi_1$ and $\mathfrak{A} \models_Z \phi_2$.

□

It is worth noting that the last step of the proof of the above theorem does not seem to work under strict semantics. It is an open question whether Theorem 13 holds under the strict semantics.

We will next show that a slight relaxation on the form of the formula immediately yields intractability of model-checking.

Theorem 14. *There exists a formula $\phi_1 \vee \phi_2$ the model-checking problem of which is* NP*-complete and such that:*

- $\phi_1 \in \mathrm{BC}(\perp, \mathrm{FO})$ *and*
- ϕ_2 *is the conjunction of two independence atoms.*

Proof. Define $\phi_1 := w \neq 1 \wedge x \perp_t y$, $\phi_2 := c_1 \perp_c c_2 \wedge x \perp_z y$. We will reduce 3-SAT to the model-checking problem of $\phi_1 \vee \phi_2$. Let $\Phi = \bigwedge_{i=1}^{n} C_i$ be a 3-SAT instance. Each $C_i = p_{i_1}$ is of the form $p_{i_2} \vee p_{i_3}$ with $p_{i_1}, p_{i_2}, p_{i_3} \in \{ v_1, \ldots, v_m, \neg v_1, \ldots, \neg v_m \}$. To this instance we associate a structure $\mathfrak{A}$ of the empty vocabulary and a team X on the variables $w, c, c_1, c_2, z, x, y, t$. The the universe of the structure $\mathfrak{A}$ is composed of $\{1, \ldots, m\}$, m new elements $a_1, \ldots, a_m$ and of $\{ v_1, \ldots, v_m, \neg v_1, \ldots, \neg v_m \} \cup \{ 0, 1 \}$. For each clause C_i we add in X the 6 assignments displayed on the left below, and for each variable v_i, we add to X the 2 assignments on the right:

w	c	c_1	c_2	z	x	y	t
0	i	1	1	i_1	p_{i_1}	p_{i_1}	a_{6i+1}
0	i	1	1	i_2	p_{i_2}	p_{i_2}	a_{6i+2}
0	i	1	1	i_3	p_{i_3}	p_{i_3}	a_{6i+3}
1	i	0	0	0	0	0	a_{6i+4}
1	i	1	0	0	0	0	a_{6i+4}
1	i	0	1	0	0	0	a_{6i+4}

w	c	c_1	c_2	z	x	y	t
0	0	0	0	i	v_i	v_i	$a_{6(n+1)+i}$
0	0	0	0	i	$\neg v_i$	$\neg v_i$	$a_{6(n+1)+i}$

We will next show that Φ is satisfiable if and only if $\mathfrak{A} \models_X \phi$.

Suppose there is an assignment $I : \{ v_1, \ldots, v_m \} \to \{0, 1\}$ that evaluates Φ to true, i.e., at least one literal in each clause is evaluated to 1. We have to split X into two sub-teams $X = Y \cup Z$ such that $\mathfrak{A} \models_Y (w \neq 1 \wedge x \perp_t y)$ and $\mathfrak{A} \models_Z (c_1 \perp_c c_2 \wedge x \perp_z y)$. We must put every assignment $s \in X$ such that $s(w) = 1$ in Z. There are exactly three such assignments per clause. We put in Z every assignment s such that $s(x) = v_i$ if $I(v_i) = 1$, and $s(x) = \neg v_i$ if $I(v_i) = 0$. The other assignments are put into Y.

For each clause C_i, one literal $p_{i_1}, p_{i_2}, p_{i_3}$ is assigned to 1 by I. Then there is at least one assignment $s(c, c_1, c_2) = (i, 1, 1)$ in Z. In Z, the assignments mapping c to i map (c, c_1, c_2) to $(i, 1, 1)$, $(i, 1, 0)$, $(i, 0, 1)$ or $(i, 0, 0)$. Thus $\mathfrak{A} \models_Z c_1 \perp_c c_2$.

If $s_1, s_2 \in Z$ are such that $s_1(z) = s_2(z) = i$, then $s_1(x)$ (analogously $s_2(x)$) is v_i if $I(v_i) = 1$, $\neg v_i$ otherwise. Therefore, $s_1(x) = s_2(x) = s_1(y) = s_2(y)$, and hence $\mathfrak{A} \models_Z x \perp_z y$ holds.

As for Y, it is immediate that $\mathfrak{A} \models_Y w \neq 1$. The only pair of assignments s_1, s_2 in X such that $s_1(t) = s_2(t)$ are $(0, 0, 0, 0, i, v_i, v_i, a_k)$ and $(0, 0, 0, 0, i, \neg v_i, \neg v_i, a_k)$, for some k. Only one of them is in Y (s_1 if $I(v_i) = 1$, s_2 otherwise). Thus $\mathfrak{A} \models_Y x \perp_t y$.

Suppose then that $X = Y \cup Z$ such that $\mathfrak{A} \models_Y (w \neq 1 \wedge x \perp_t y)$ and $\mathfrak{A} \models_Z (c_1 \perp_c c_2 \wedge x \perp_z y)$. Define an assignment I of the variables of Φ by: $I(v_i) = 1$ if $s_i^t := (0, 0, 0, 0, i, v_i, v_i, a_k)$ is in Z, $I(v_i) = 0$ if $s_i^f := (0, 0, 0, 0, i, \neg v_i, \neg v_i, a_k)$ is in Z. Since $\mathfrak{A} \models_Z x \perp_z y$, $s_i^f(z) = s_i^t(z)$ and because there is no $s' \in X$ such that $s'(x) = s_i^f(x), s'(y) = s_i^t(y)$ and $s'(z) = s_i^f(z)$, for each i at most one of s_i^f, s_i^t can be in Z. Similarly, because $\mathfrak{A} \models_Y x \perp_t y$, only one of them can be in Y. Thus I is indeed a function.

Since $\mathfrak{A} \models_Z x \perp_z y$ and there is no assignment in X such that $(x, y) \mapsto (v_i, \neg v_i)$, every pair $s_1, s_2 \in Z$ such that $s_1(z) = s_2(z) = i$ must have the same value of x and y. Every assignment representing a clause in Z respects the choice of I. Furthermore, since $\mathfrak{A} \models_Z c_1 \perp_c c_2$ and $(w, c, c_1, c_2) \mapsto (1, i, 0, 0), (1, i, 1, 0), (1, i, 0, 1)$ are in Z, $(1, i, 1, 1)$ must be in Z, i.e., at least one assignment per clause is in Z. By the above we may conclude that I satisfies Φ: at least one literal per clause is evaluate to 1 by I.

Finally it is not difficult to check that the splitting of X into Y and Z, if possible, can be realized with disjoint Y and Z. Hence the hardness holds under both strict and lax semantics

□

It is worth noting that the formula in the previous result has disjunction-width two whereas the formula in Corollary 9 simulating dependence atoms by conditional independence atoms has width three. We will next show an analog of these results for pure (i.e., non-conditional) independence atoms.

Theorem 15. *The model checking problem is* NP*-complete for*

$$\phi := (x \perp y) \vee (x \perp y) \vee (x \perp y) \vee x \neq y.$$

Proof. Membership in NP is obvious. Hardness is proved by reduction from the following NP-complete problem 3-clique cover (see [6]): Given $G = (V_G, E_G)$ a finite graph, decide if there exists a collection of disjoint triangle subgraphs of G, that cover the vertex set V_G of G.

So, let $G = (V_G, E_G)$ be a finite graph, $\mathfrak{A} = V_G$ be a first-order structure of the empty signature, and X the team

$$X = \{ (v, v) \,:\, v \in V_G \} \cup \{ (v_1, v_2), (v_2, v_1) \,:\, (v_1, v_2) \in E_G \},$$

where (v_1, v_2) denotes the assignment s with $s(x) = v_1$ and $s(y) = v_2$. We are going to show that G has a 3-clique cover if and only if $\mathfrak{A} \models_X \phi$.

x	y
v_1	v_1
v_2	v_2
$\vdots$	$\vdots$
$v_{\lvert V_G \rvert}$	$v_{\lvert V_G \rvert}$
v_1	v_2
v_2	v_1
v_2	v_4
v_4	v_2
$\vdots$	$\vdots$

Suppose that G has a 3-clique cover, i.e., there exists C_1, C_2, C_3 three cliques such that $V_G = V_{C_1} \cup V_{C_2} \cup V_{C_3}$. We have to prove $\mathfrak{A} \models_X \phi$. For $i \in \{ 1, 2, 3 \}$, let

$$X_i = \{ (v, v) \,:\, v \in C_i \} \cup \{ (v_1, v_2), (v_2, v_1) \mid v_1, v_2 \in C_i \}$$

and $X_4 = X \setminus (X_1 \cup X_2 \cup X_3)$.

Because it is a vertex cover, every assignment of the form (v, v) is contained in $X_1 \cup X_2 \cup X_3$ and not in X_4, i.e. $\mathfrak{A} \models_{X_4} x \neq y$.

Let $i \in \{ 1, 2, 3 \}$ and $s, s' \in X_i$ be two assignments. If $s(x, y) = (v, v)$ and $s'(x, y) = (v', v')$, then $v, v' \in C_i$ and there exists two assignments s_1, s_2 in X_i such that $s_1(x, y) = (v, v')$ and $s_2(x, y) = (v', v)$ by construction. Similarly if $s(x, y) = (v, v)$ and $s'(x, y) = (v_1, v_2)$, there exists in X_i the assignments (v, v_2) and (v_1, v) (even if $v_1 = v$ or $v_2 = v$). Finally, if $s(x, y) = (v_1, v_2)$ and $s'(x, y) = (v_1', v_2')$, the assignments (v_1, v_1), (v_2, v_2), (v_1', v_1'), (v_2', v_2') are in X_i and so are (v_1, v_2'), (v_1', v_2). The above implies that $\mathfrak{A} \models_{X_i} x \perp y$.

For the converse implication, suppose that $\mathfrak{A} \models_X (x \perp y) \vee (x \perp y) \vee (x \perp y) \vee x \neq y$, then $X = X_1 \cup X_2 \cup X_3 \cup X_4$ such that $\mathfrak{A} \models_{X_i} x \perp y$ for $i \in \{1,2,3\}$ and $\mathfrak{A} \models_{X_4} x \neq y$. Let C_i, for $i \in \{1,2,3\}$, be the graph whose vertices are $\{v : (v,v) \in X_i\}$ and edges are

$$\{(v_1,v_2) : (v_1,v_2) \in X_i \text{ and } (v_2,v_1) \in X_i\}.$$

Note that some C_i can be empty but they form a vertex cover of G as no assignment (v,v) is in X_4. If $v, v' \in C_i$ then (v,v) and (v',v') are in X_i. By independence, (v,v') and (v',v) are also in X_i. Therefore the edge (v,v') is in C_i: C_i is a clique. Therefore, G is covered by the three disjoint cliques $C'_1 = C_1$, $C'_2 = C_2 \setminus C_1$ and $C'_3 = C_3 \setminus (C_1 \cup C_2)$.

If we are in the strict semantics, the sets X_i are disjoint and so are the cliques. If we are in lax semantic, G is covered by the disjoint cliques $C'_1 = C_1$, $C'_2 = C_2 \setminus C_1$ and $C'_3 = C_3 \setminus (C_1 \cup C_2)$. In both cases, G has a 3-clique cover.

□

3.2 The case of quantified formulas

In this section we show that existential quantification even without disjunction and even in the empty vocabulary makes the model checking problem hard for both dependence and independence logic.

Theorem 16. *There exists a formula ϕ with* NP*-complete model-checking problem where $\phi \equiv \exists x \psi$ and ψ is a conjunction of two dependence atoms.*

Proof. Before giving the proof, let us consider the following problem: Given a graph G with n^2 vertices, are the vertices of G colourable with n colors (such that no adjacent vertices carries the same color). This problem is easily seen to be a generalization of the well-known NP-complete 3-coloring problem (see [6]). Indeed, let $G = (V_0, E_0)$ be an undirected graph with $|V_0| = n$ vertices, let $K_{n-3} = (V_1, E_1)$ be the complete graph with $|V_1| = n-3$ vertices (hence $|E_1| = (n-3)(n-4)/2$ edges) and let G' be the graph with vertex set $V' = V_0 \cup V_1 \cup V_2$ with V_2 of size $n^2 - n - n + 3$ (hence $|V'| = n^2$) and edge set $E_0 \cup E_1 \cup \{\{x,y\} : x \in V_0, y \in V_1\}$. Then, it is easy to see that G is 3-colorable iff G' is n-colorable.

Let us now define the formula ϕ as follows:

$$\phi \equiv \exists x (=(x, r_1, r_2, e, m) \wedge =(v_1, v_2, x)).$$

We will reduce the problem of determining whether a graph G with n^2 vertices is n-colorable to the model-checking problem of ϕ.

Let G be a graph with n^2 vertices $V_G = \{\alpha_0, \ldots, \alpha_{n^2-1}\}$, $\mathfrak{A} = \{0, \ldots, n-1\}$ a first order structure of the empty signature and $X = \{s_i^j \mid i \in \{0, \ldots, n^2-1\},\ 0 \leq j \leq i\}$ be a team such that :

- $s_i^j(v_1) = \lfloor i/n \rfloor$ and $s_i^j(v_2) = i \mod n$. In other words, $s_i^j(v_1, v_2)$ is the decomposition of i in base n.
- $s_i^j(r_1) = \lfloor j/n \rfloor$ and $s_i^j(r_2) = j \mod n$. In other words, $s_i^j(r_1, r_2)$ is the decomposition of j in base n.
- $s_i^j(m) = 0$ if $i \neq j$ and $s_i^i(m) = 1$.
- $s_j^i(e) = 1$, if $j = i$, or if there is an edge between α_i and α_j with $j > i$. Otherwise $s_j^i(e) = 0$.

For example, for $n = 2$ and $E_G = \{\{0,1\}, \{1,2\}, \{0,2\}, \{2,3\}\}$, we obtain the following team on the universe $A = \{0,1\}$:

x	v_1	v_2	r_1	r_2	m	e
	0	0	0	0	1	1
	0	1	0	0	0	1
	0	1	0	1	1	1
	1	0	0	0	0	1
	1	0	0	1	0	1
	1	0	1	0	1	1
	1	1	0	0	0	0
	1	1	0	1	0	0
	1	1	1	0	0	1
	1	1	1	1	1	1

G is n-colourable iff $\mathfrak{A} \models_X \phi$

We are going to demonstrate that $\mathfrak{A} \models_X \phi$ if and only if G is n-colourable.

First the left to right implication. Since $\mathfrak{A} \models_X \phi$ there exists a mapping $F\colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X(F/x)} =(x, r_1, r_2, e, m) \wedge =(v_1, v_2, x)$. By downwards closure, we may assume without loss of generality that $F(s)$ is a singleton for all $s \in X$. Since $=(v_1, v_2, x)$ holds, F induces a mapping $F'\colon V_G \to A$, by $F'(\alpha_i) := a$, for the unique a such that $F(s_i^0) = \{a\}$. If there is an edge between α_i and $\alpha_{i'}$, $i' > i$, then $s_{i'}^i(e) = 1 = s_i^i(e)$. Furthermore, $s_{i'}^i(r_1, r_2) = s_i^i(r_1, r_2)$ but $s_{i'}^i(m) = 0$ and $s_i^i(m) = 1$. Therefore, because the atom $=(x, r_1, r_2, e, m)$ holds, we must have $F(s_i^i) \neq F(s_{i'}^i)$. Thus $F'(\alpha_i) \neq F'(\alpha_{i'})$ if there is an edge between α_i and $\alpha_{i'}$. This shows that F' is a colouring of G with $|A| = n$ colours.

Let us then consider the right to left implication. Let $c : V_G \to \{0, \ldots, n-1\}$ be an n colouring. We extend X to variable x with a new team X' such that $s_i^j(x) = c(\alpha_i)$. The value of x depends only on i, which is encoded in (v_1, v_2), i.e., $\mathfrak{A} \models_{X'} =(v_1, v_2, x)$.

Let $s_i^j, s_{i'}^{j'}$ be two assignments of X'. Suppose that $s_i^j(r_1, r_2, e) = s_{i'}^{j'}(r_1, r_2, e)$ but $s_i^j(m) \neq s_{i'}^{j'}(m)$. In this case we must check that $s_i^j(x)$ is different from $s_{i'}^{j'}(x)$ (because $\mathfrak{A} \models_{X'} =(x, r_1, r_2, e, m)$). Now it holds that $j = j'$ because $s_i^j(r_1, r_2) = s_{i'}^{j'}(r_1, r_2)$. Furthermore, since $s_i^j(m) \neq s_{i'}^{j'}(m)$, either $i = j$ or $i' = j'$. Let us suppose $i = j$. Because $1 = s_i^i(e) = s_i^j(e) = s_{i'}^{j'}(e) = s_{i'}^i(e)$, there is an edge between α_i and $\alpha_{i'}$ in G. Therefore $c(i) \neq c(i')$ and $s_i^j(x) \neq s_{i'}^{j'}(x)$. □

By encoding dependence atoms in terms of conditional independence atoms we get the analogous results for free for independence logic.

Corollary 17. *There is a formula ϕ of independence logic of empty non-logical vocabulary built with $\exists$ and $\wedge$ whose model-checking problem is* NP*-complete.*

We will next show that this corollary can be strengthened in the case of independence logic under the strict semantics. In other words, we will now show a version of Theorem 16 for pure independence atoms under the strict semantics. Let $\psi(t, c, v)$ be the following formula over signature $\sigma = \{R\}$, where R is a ternary relation symbol and t, c and v are free variables:

$$\psi(t, c, v) := \exists x(t \perp x \wedge R(c, v, x)).$$

Proposition 18. *For all propositional formulas ϕ in 3-*CNF*, one can compute in polynomial time a team X, with* $\mathrm{Dom}(X) = \{t, c, v\}$ *and a structure $\mathfrak{A}$ such that:*

$$\phi \text{ is satisfiable} \iff \mathfrak{A} \models_X \psi(t, c, v),$$

under the strict semantics.

Proof. Without loss of generality, let $\phi = \bigwedge_{i=1}^{m} C_i$ be a 3-CNF formula over a set $V = \{v_1, \ldots, v_n\}$ of variables of size n. Let $C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$, with $l_{i_j} \in \{v_1, ..., v_n, \neg v_1, ..., \neg v_n\}$. We first describe the relation R built on the domain $D = \{0, \ldots, m+n, v_1, ..., v_n, \neg v_1, ..., \neg v_n\}$:

$$\begin{aligned} R = \ & \{(0, i, v_i), (0, i, \neg v_i) : 1 \leq i \leq n\} \cup \\ & \{(i, 0, l_{i_1}), (i, 0, l_{i_2}), (i, 0, l_{i_3}) : 1 \leq i \leq m,\ C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}\} \cup \\ & \{(m+i, 0, v_i), (m+i, 0, \neg v_i) : 1 \leq i \leq n\}. \end{aligned}$$

Let $\mathfrak{A} = (D, R)$. Finally, team X is the union of the two assignment sets Y and Z below:

Y:

c	v	t
0	1	0
0	2	0
⋮	⋮	⋮
0	n	0

Z:

c	v	t
1	0	1
2	0	1
⋮	⋮	⋮
m	0	1
$m+1$	0	1
⋮	⋮	⋮
$m+n$	0	1

Variable t encodes the type of the object in consideration: 0 for a clause, 1 for a Boolean variable. The first n assignments deal with variables (hence the value of c is set to 0, by convention), the last m assignments deal with clauses (hence, v is set to 0). It now remains to show that

$$\phi \text{ is satisfiable} \iff \mathfrak{A} \models_X \psi(t, c, v),$$

where $\psi(t, c, v)$ is the formula $\exists x\ (t \perp x \wedge R(c, v, x))$. Suppose first that ϕ is satisfiable and let $I : V \to \{0, 1\}$ be such that $I(\phi) = 1$. Let $F : X \to D$ be such that:

(1) if $s(c) = 0$ and $s(v) = i$, then $F(s) = v_i$ if $I(v_i) = 1$ and $F(s) = \neg v_i$ if $I(v_i) = 0$. Similarly, if $s(c) = m + i$, $i \geq 1$, then $F(s) = v_i$ if $I(v_i) = 1$ and $F(s) = \neg v_i$ if $I(v_i) = 0$.
(2) if $s(c) = i$, $1 \leq i \leq m$, then $F(s) = l_{i_j}$, for $j \leq 3$, such that $I(l_{i_j}) = 1$. Such a j always exists since $I \models \phi$.

Let $X' = X(F/x)$. It is clear that for all $s \in X'$, $\mathfrak{A} \models_s R(c, v, x)$ holds by the construction. In X', variable t takes only two values 0 and 1. Let now

$$\begin{aligned} V_0 &= \{s(x) : s \in X' \text{ and } s(t) = 0\}, \\ V_1 &= \{s(x) : s \in X' \text{ and } s(t) = 1\}. \end{aligned}$$

Now to show the claim $\mathfrak{A} \models_X \psi(t, c, v)$, it remains to show $\mathfrak{A} \models_{X'} t \perp x$. For this it suffices to show that $V_0 = V_1$. Note that, for all $i \leq n$, either v_i or $\neg v_i$ belongs to V_0. Also, by the construction, $V_0 \subseteq V_1$. Indeed, by item (1), for any s with $s(t) = s(c) = 0$ and $s(v) = i$ it holds that $s(x) = s'(x)$, where $s'(c) = m + i$ and $s'(t) = 1$. Suppose now that there exists $s \in X'$ such that $s(x) \in V_1$ and $s(x) \notin V_0$. Clearly, for such an s, $s(c) = i$, for some $1 \leq i \leq m$. Then, by the

construction of the function F, $s(x) = l_{i_j}$, for $j \leq 3$, such that $I(l_{i_j}) = 1$. But then again by the definition of F, $s(x) \in V_0$ which is a contradiction. Therefore, $V_0 = V_1$, $\mathfrak{A} \models_{X'} t \perp x$, and hence $\mathfrak{A} \models_X \psi(t, c, v)$.

We now prove the other implication. Suppose $\mathfrak{A} \models_X \psi(t, c, v)$ and let $X' = X(F/x)$ be such that

$$\mathfrak{A} \models_{X'} t \perp x \wedge R(c, v, x).$$

Because $\mathfrak{A} \models_{X'} t \perp x$ and $s(t) \in \{0, 1\}$ for all $s \in X'$, it holds that $V_0 = V_1$, where V_0 and V_1 are as defined in the first part of the proof above. Together with the definition of the relation R, this implies that if $s, s' \in X$ such that $s(c) = 0, s(v) = i$ and $s'(c) = m + i, s'(v) = 0$, then $s(x) = s'(x) \in \{v_i, \neg v_i\}$. Define now the Boolean assignment $I : V \to \{0, 1\}$ by

$$I(v_i) = 1 \iff s(x) = v_i \text{ for } s \text{ s.t. } s(c) = 0, s(v) = i.$$

Now consider $s \in X'$ with $s(c) = i \in \{1, ..., m\}$. By definition of R, $s(x) = l_{i_j}$ (for some $j \leq 3$). Since $V_1 \subseteq V_0$ there must exists $s' \in X'$ s.t. $s'(t) = 0, s'(x) = l_{i_j}$ and, also immediately by the construction of X, $s'(c) = 0$ and $s'(v) = i$. The definition of I implies that $I(l_{i_j}) = 1$ and that clause C_i is satisfied. □

We end this section by noting that existential quantifiers cannot be replaced by universal quantifiers in the above theorems.

Proposition 19. *The model-checking problem for formulas of dependence or independence logic using only universal quantification and conjunction is in* L.

Proof. Given ϕ, we first transform it into prenex normal-form exactly as in first-order logic [25]. We may hence assume that ϕ has the form

$$\forall x_1 \ldots \forall x_n \bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m),$$

where θ_i is either a first-order, dependence, or independence atom. Let $\mathfrak{A}$ be a model, and X be a team of A with domain $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$. As in [25], the formula $\bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m)$ can be expressed by a first-order sentence ψ when the team X is represented by the $n + m$-ary relation $X(\overline{x}, \overline{y})$, that is,

$$\mathfrak{A} \models_X \bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m) \Leftrightarrow (\mathfrak{A}, X(\overline{x}, \overline{y})) \models \psi.$$

Since $X(\overline{x}, \overline{y})$ is a first-order definable extension of $X(\overline{y})$ it is clear that we can construct a FO-sentence ψ' such that

$$\mathfrak{A} \models_X \forall \overline{x} \bigwedge \theta_i(\overline{x}, \overline{y}) \Leftrightarrow (\mathfrak{A}, X(\overline{y})) \models \psi',$$

holds for all structures $\mathfrak{A}$ and teams X with domain $\{y_1, \ldots, y_m\}$. The claim follows from the fact that the data complexity of FO is in L. □

4 INCLUSION LOGIC UNDER THE LAX SEMANTICS

Recall that a CNF formula Φ is called dual-Horn if each of its clauses contains at most one negative literal. The satisfiability problem of dual-Horn CNF formulas, DUAL-HORN-SAT, is known to be PTIME-complete (see [6]).

In this section we show that the model-checking problem of inclusion logic formulas under the lax semantics can be reduced to DUAL-HORN-SAT.

For a team X, $\overline{x} = \langle x_{i_1}, ..., x_{i_n} \rangle \in \mathrm{dom}(X)^n$, and $s \in X$, we denote by $s(\overline{x})$ the restriction of s to the variables $x_{i_1}, ..., x_{i_n}$. In this section, σ denotes a relational signature.

Proposition 20. *There exists an algorithm which, given $\phi \in \mathrm{FO}(\subseteq)$, a structure $\mathfrak{A}$ over σ, and a team X such that $\mathrm{Var}(\phi) \subseteq \mathrm{dom}(X)$, outputs a propositional formula Ψ in dual-Horn form such that: $\mathfrak{A} \models_X \phi \iff \Psi$ is satisfiable. Furthermore, when ϕ is fixed, the algorithm runs in logarithmic space in the size of $\mathfrak{A}$ and X.*

Proof. Let $\phi, \mathfrak{A}, X$ be as above. For any team X, we will consider the set $\mathfrak{X}$ of propositional variables $X[s]$ for $s \in A^{\mathrm{dom}(X)}$. Starting from ϕ, $\mathfrak{A}$, and X we decompose step by step the formula ϕ into subformulas (until reaching its atomic subformulas) and different teams Y, Z, ... and control the relationships between the different teams by propositional dual-Horn formulas built over the propositional variables issued from $X, Y, Z, \ldots$. Let $\mathcal{S} = \{(\phi, X, V)\}$, where $V = \mathrm{dom}(X)$, and $C = \{X[s] : s \in X\} \cup \{\neg X[s] : s \notin X\}$. The propositional formula Ψ is now constructed inductively as follows.

As long as $\mathcal{S} \neq \emptyset$, we apply the following rule: Pick (ϕ, X, V) in $\mathcal{S}$ and apply the following rules.

- If ϕ is $R(\overline{x})$ with R a literal of σ then: $\mathcal{S} := \mathcal{S} \backslash \{(\phi, X, V)\}$ and $C := C \cup \{X[s] \to \top : s \in A^V$ and $\mathfrak{A} \models_X R(\overline{x})\} \cup \{X[s] \to \bot : s \in A^V$ and $\mathfrak{A} \not\models_X R(\overline{x})\}$. Clearly, it holds that $\mathfrak{A} \models_X R(\overline{x})$ iff $\bigwedge C$ is satisfiable.
- If ϕ is $\overline{x} \subseteq \overline{y}$ then: $\mathcal{S} := \mathcal{S} \backslash \{(\phi, X, V)\}$ and

$$C := C \cup \{X[s] \to \bigvee_{s' \in A^V, s'(\overline{y}) = s(\overline{x})} X[s'] : s \in A^V\}.$$

  It holds that $\mathfrak{A} \models_X \overline{x} \subseteq \overline{y}$ iff $\bigwedge_{s \in A^V} (X[s] \to \bigvee_{s' \in A^V, s'(\overline{y}) = s(\overline{x})} X[s'])$ is satisfiable.
- If ϕ is $\exists x \psi$, then: $\mathcal{S} := (\mathcal{S} \backslash \{(\phi, X, V)\}) \cup \{(\psi, Y, V \cup \{x\})\}$ and

$$C := C \cup \{X[s] \to \bigvee_{a \in A} Y[s(a/x)] : s \in A^V\},$$

  where the $Y[s]$, $s \in A^{V \cup \{x\}}$ are new propositional variables (not used in C). If $\mathfrak{A} \models_X \exists x \psi$ then, there exists a function $F : X \to \mathcal{P}(A) \setminus \{\emptyset\}$, such that $\mathfrak{A} \models_{X(F/x)} \psi$. In other words, $\mathfrak{A} \models_Y \psi$ for some team Y defined by the solutions of the constraint $\bigwedge_{s \in A^V} X[s] \to \bigvee_{a \in A} Y[s(a/x)]$ (which define a suitable function F). Conversely, if $\mathfrak{A} \models_Y \psi$ for a team Y as above defined from X, then clearly $\mathfrak{A} \models_X \exists x \psi$.
- If ϕ is $\forall x \psi$, then: $\mathcal{S} := (\mathcal{S} \backslash \{(\phi, X, V)\}) \cup \{(\psi, Y, V \cup \{x\})\}$ and

$$C := C \cup \{X[s] \to Y[s'] : s \in A^V, s' \in A^{V \cup \{x\}} \text{ s.t. } s'(\overline{x}) = s(\overline{x})\},$$

  where the $Y[s]$, $s \in A^{V \cup \{x\}}$ are new propositional variables (not used in C). The conclusion is similar as for the preceding case.
- If ϕ is $\psi_1 \wedge \psi_2$ then: $\mathcal{S} := (\mathcal{S} \backslash \{(\phi, X, V)\}) \cup \{(\psi_1, X, V), (\psi_2, X, V)\}$ and C is unchanged. By definition, $\mathfrak{A} \models_X \phi$ iff $\mathfrak{A} \models_X \psi_1 \wedge \psi_2$.
- If ϕ is $\psi_1 \vee \psi_2$ then: $\mathcal{S} := (\mathcal{S} \backslash \{(\phi, X, V)\}) \cup \{(\psi_1, Y, V), (\psi_2, Z, V)\}$ and

$$C := C \cup \{X[s] \to Y[s] \vee Z[s] : s \in A^V\} \cup \{Y[s] \to X[s], Z[s] \to X[s] : s \in A^V\}$$

  where again the $Y[s]$ and $Z[s]$, $s \in A^V$ are new propositional variables (not used in C). Here again, $\mathfrak{A} \models_X \phi$ if and only if $\mathfrak{A} \models_Y \psi_1$ and $\mathfrak{A} \models_Z \psi_2$ for some suitable Y and Z such that $Y \cup Z = X$ which is exactly what is stated by the Boolean constraints.

Observe that each new clause added to C during the process is of dual-Horn form, i.e., contains at most one negative literal. Observe also, that applied to some (ϕ, X, r), the algorithm above only adds triples $\mathcal{S}$ whose first component is a proper subformula of ϕ and eliminates (ϕ, X, r). When the formula ϕ is atomic, no new triple is added afterwards.

Hence the algorithm will eventually terminate with $\mathcal{S} = \emptyset$. Setting $\Psi := \bigwedge_{C \in \mathcal{C}} C$, it can easily be proved by induction that: $\mathfrak{A} \models_X \phi$ iff Ψ is satisfiable.

Observe also that each clause in $\mathcal{C}$ can be constructed from X and $\mathfrak{A}$ by simply running through their elements (using their index) hence in logarithmic space. □

Remark 1. The construction of Proposition 20 can be done in principle for any kind of atom: dependence, independence, exclusion, constancy etc. To illustrate this remark, one could translate in the above proof a dependence atom of the form $=(\overline{x}, y)$ by (using the notations of the proof):

$$\bigwedge_{\substack{s,s' \in A^r \\ s(\overline{x})=s'(\overline{x}) \wedge s(y) \neq s'(y)}} (\neg X[s] \vee \neg X[s']).$$

The additional clauses are of length two. A similar treatment can be done for independence atoms $\overline{x} \perp_{\overline{y}} \overline{z}$. In the two cases however, the resulting formula is not in Dual-Horn form anymore and there is no way to do so (unless PTIME = NP).

Since deciding the satisfiability of a propositional formula in dual-Horn form can be done in polynomial time we obtain the following already known ([5]) corollary.

Corollary 21. *The data complexity of* FO($\subseteq$) *under the lax semantics is in* PTIME.

5 INCLUSION LOGIC UNDER THE STRICT SEMANTICS

In this section we consider model-checking of inclusion logic formulas under the strict semantics. By the result of [4], inclusion logic with the strict semantics is equi-expressive with dependence logic. The following theorem shows that NP-completeness can be attained with quite simple formulas as in Theorem 16 combining strict existential quantification and inclusions atoms.

Proposition 22. *Let $\psi(c, v)$ be the following formula over signature $\sigma = \{R\}$:*

$$\psi(c, v) = \exists x \exists y (y \subseteq x \wedge R(c, v, x, y)).$$

For all propositional formulas ϕ in 3-CNF*, one can compute in polynomial time a team X with domain $\{c, v\}$ and a structure $\mathfrak{A}$ such that: ϕ is satisfiable $\Leftrightarrow \mathfrak{A} \models_X \psi(c, v)$ under the strict semantics.*

Proof. Let $\phi = \bigwedge_{i=1}^{m} C_i$ be a 3-CNF formula over a set $V = \{v_1, \ldots, v_n\}$ of variables. Let $C_i = \ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}$, with $\ell_{i_j} \in \{v_1, \ldots, v_n, \neg v_1, \ldots, \neg v_n\}$.

The domain of the structure $\mathfrak{A}$ is $A = \{0, \ldots, n, v_1, \ldots, v_n, \neg v_1, \ldots, \neg v_n\}$. The relation R in this structure is:

$$\begin{aligned} R = &\{(0, i, v_i, 0), (0, i, \neg v_i, 0) | 1 \leq 1 \leq n\} \cup \\ &\{(i, 0, 0, \ell_{i_1}), (i, 0, 0, \ell_{i_2}), (i, 0, 0, \ell_{i_3}) | 1 \leq i \leq m\} \end{aligned}$$

Finally, the team X isgiven by the following table:

$$X = \begin{array}{|c|c|}\hline c & v \\ \hline 0 & 1 \\ \hline \vdots & \vdots \\ \hline 0 & n \\ \hline 1 & 0 \\ \hline \vdots & \vdots \\ \hline m & 0 \\ \hline \end{array}$$

Now we claim that ϕ is satisfiable, if and only if $\mathfrak{A} \models_X \exists x \exists y (y \subseteq x \wedge R(c, v, x, y))$.

Let us suppose that $\mathfrak{A} \models_X \exists x \exists y (y \subseteq x \wedge R(c, v, x, y))$. Then there is an extension X' of X to the variables x and y such that $\mathfrak{A} \models_{X'} x \subseteq y \wedge R(c, v, x, y)$. Note that X' has, e.g., the following shape:

$$X' : \begin{array}{|c|c|c|c|}\hline c & v & x & y \\ \hline 0 & 1 & \neg v_1 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline 0 & i & \neg v_i & 0 \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline 0 & n & v_n & 0 \\ \hline 1 & 0 & 0 & \ell_{1_2} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline i & 0 & 0 & \ell_{i_1} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline m & 0 & 0 & \ell_{m_2} \\ \hline \end{array}$$

Note that for each row of X the variables x and y get exactly one value in X'.

Let $I : V \to \{0, 1\}$ be the following assignment of the variables of ϕ: let $s \in X$ such that $s(v) = i$; we set $I(v_i) = 1$ if $s(x) = v_i$, $I(v_i) = 0$ otherwise (i.e., if $s(x) = \neg v_i$). Because $\mathfrak{A} \models_{X'} R(c, v, x, y)$ such s exists and $s(x) \in \{v_i, \neg v_i\}$. Furthermore, s is the only assignment in X' such that $s(v) = i$ so there is no ambiguity in the definition of I.

Now we have to check that for every clause of ϕ there is a literal which is evaluated to 1 by I. Let C_i be a clause of ϕ and s the element of X' such that $s(c) = i$. Since $s(y)$ is a literal of C_i and $\mathfrak{A} \models_{X'} y \subseteq x$, there exists $s' \in X'$ such that $s'(x) = s(y)$. But $I(s(x)) = 1$ by definition, thus a literal of C_i is evaluated to 1 by I and hence I satisfies ϕ.

Suppose then that ϕ is satisfiable and let $I : V \to \{0, 1\}$ an assignment of the variables of ϕ which satisfies ϕ. We extend X to a team X' over variables $\{c, v, x, y\}$ as follows: for $1 \leq i \leq n$, if $s \in X$ is such that $s(v) = i$ we set $s(x) = v_i$ if $I(v_i) = 1$, and $s(x) = \neg v_i$ otherwise. Furthermore, $s(y) = 0$.

For $1 \leq i \leq m$, if $s \in X$ is such that $s(c) = i$, we set $s(x) = 0$ and $s(y) = \ell_{i_j}$ where ℓ_{i_j} is a literal of C_i which is evaluated to 1 by I. It is now easy to check that $\mathfrak{A} \models_{X'} y \subseteq x \wedge R(c, v, x, y)$, and hence $\mathfrak{A} \models_X \exists x \exists y (y \subseteq x) \wedge R(c, v, x, y)$. □

The next proposition shows that NP-completeness can be also attained by combining strict disjunction with inclusions atoms.

Proposition 23. *There exists formulas ϕ_1, ϕ_2, ϕ_3 built with $\subseteq, \wedge$ such that the model checking problem for $\phi_1 \vee \phi_2 \vee \phi_3$ under the strict semantics is NP-complete.*

Proof. Membership in NP is obvious. For hardness, we exhibit a polynomial time reduction to 1-in-3-sat.

Let ψ be the following formula over variables $\{l, v, c, a, b, w_1, w_2, w_3\}$:

$$\begin{aligned}\psi \equiv \quad & (\ell \subseteq v \wedge b \subseteq c \wedge a \subseteq w_1) \vee \\ & (b \subseteq c \wedge a \subseteq w_2) \vee \\ & (b \subseteq c \wedge a \subseteq w_3))\end{aligned}$$

We will show that for all positive 3-cnf formulas ϕ, one can compute in polynomial time a team X and a structure $\mathfrak{A}$ such that:

$$\phi \text{ is an instance of 1-in-3-sat } \Leftrightarrow \mathfrak{A} \models_X \psi(l, v, c, a, b, w_1, w_2, w_3)$$

Let $\phi = \bigwedge_{i=1}^{m} C_i$ be a positive 3-cnf formula over a set $V = \{v_1, \ldots, v_n\}$ of variables. Let $C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$. Recall that ϕ is an instance of the problem 1-in-3-SAT if and only if there is a truth assignment such that each clause of ϕ has exactly one true variable.

The domain of the structure $\mathfrak{A}$ is $D = \{v_1, \ldots, v_n, 0, \ldots, m\}$. The team X is $Y \cup Z$, see Table 1.

Now we claim that ϕ is an instance of 1-in-3-sat, if and only if $\mathfrak{A} \models_X \psi$.

Let us suppose that $\mathfrak{A} \models_X \psi$, i.e., there exists a partition of X into three subsets X_1, X_2 and X_3 such that

$$\begin{aligned}&\mathfrak{A} \models_{X_1} b \subseteq c \wedge a \subseteq w_1, \\ &\mathfrak{A} \models_{X_2} b \subseteq c \wedge a \subseteq w_2, \\ &\mathfrak{A} \models_{X_3} b \subseteq c \wedge a \subseteq w_3, \\ &\mathfrak{A} \models_{X_1} \ell \subseteq v.\end{aligned}$$

We will define an assignment I over V witnessing that ϕ is an instance of 1-in-3-sat: for $1 \leq i \leq n$ there exists a unique $s \in X$ such that $s(v) = v_i$. If $s \in X_1$, we set $I(v_i) = 1$ and $I(v_i) = 0$ otherwise. As s is unique and the sets X_i are disjoint (because we are in the strict semantics), I is well defined.

We have to check that for any $1 \leq i \leq m$, exactly one of the variables of clause C_i is evaluated to 1 by the assignment I.

Now $\mathfrak{A} \models_{X_1} a \subseteq w_1$, $\mathfrak{A} \models_{X_2} a \subseteq w_2$ and $\mathfrak{A} \models_{X_3} a \subseteq w_3$ imply that every assignment $s \in X$ such that $s(a) = 1$ must be in X_1. Therefore $X_1(b) = \{0, 1, \ldots, m\}$. Similarly $X_2(b) = X_3(b) = \{0, 1, \ldots, m\}$.

The variable c stores the index of a clause. Because $\mathfrak{A} \models_{X_1} b \subseteq c$, every clause C_i has an assignment $s \in X_1$ such that $s(c) = i$, and the same holds for the sets X_2 and in X_3. Thus the claim follows.

Let $I : \{v_1, \ldots, v_n\} \to \{0, 1\}$ be an assignment witnessing that ϕ is an instance of 1-in-3-sat. Define a partition of X into X_1, X_2, X_3 as follows. Let $s \in X$.

(1) If $s(v) = v_i$ and $I(v_i) = 1$, we assign $s \in X_1$. If $s(v) = v_i$ and $I(v_i) = 0$ we assign $s \in X_2$,
(2) For every $1 \leq i \leq m$, $C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$. Let $s \in X$ be such that $s(c) = i$. We send the unique s such that $s(\ell) = l_{i_j}$ and $I(l_j) = 1$ to X_1 and assign exactly one of the remaining two such assignments s to X_2 and X_3,
(3) If $s(a) = k$, we send s to X_k.

By (1) and (2) it now clearly holds that $\mathfrak{A} \models_{X_1} \ell \subseteq v$. Furthermore, by (2) and (3) it holds that $\mathfrak{A} \models_{X_k} b \subseteq c$ and $\mathfrak{A} \models_{X_k} a \subseteq w_k$, for $k = 1, 2, 3$, respectively.

Y :

ℓ	v	c	a	b	w_1	w_2	w_3
0	v_1	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	v_n	0	0	0	0	0	0
ℓ_{1_1}	0	1	0	0	0	0	0
ℓ_{1_2}	0	1	0	0	0	0	0
ℓ_{1_3}	0	1	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ℓ_{i_1}	0	i	0	0	0	0	0
ℓ_{i_2}	0	i	0	0	0	0	0
ℓ_{i_3}	0	i	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ℓ_{m_1}	0	m	0	0	0	0	0
ℓ_{m_2}	0	m	0	0	0	0	0
ℓ_{m_3}	0	m	0	0	0	0	0

Z :

ℓ	v	c	a	b	w_1	w_2	w_3
0	0	0	1	1	1	2	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	1	m	1	2	3
0	0	0	2	1	1	2	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	2	m	1	2	3
0	0	0	3	1	1	2	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	3	m	1	2	3

Table 1

□

6 CONCLUSION

On this paper we have studied the tractability/intractability frontier of data complexity of both quantifier-free and quantified dependence, independence, and inclusion logic formulas. Furthermore, we defined a novel translation of inclusion logic formulas into dual-Horn propositional formulas, and used it to show that the data-complexity of inclusion logic is in PTIME. In a paper under preparation we shall consider similar questions for quantifier-free formulas containing, in addition to dependence and independence atoms, also so-called anonymity atoms. Although our results shed light on the tractability/intractability frontiers studied in this article, many open questions related to the data-complexity of quantifier-free independence logic formulas remain. The general goal is to find fragments where we can prove PTIME/NP-complete dichotomy results. Our results on disjunctions of independence atoms show that the

data-complexity question is more complex than merely the length of the disjunction, as is the case with dependence atoms. In a different direction, it is an open question whether Theorem 13 holds under the strict semantics.

ACKNOWLEDGEMENTS

The second author was supported by the Academy of Finland grant 308712. The fourth author was supported by the Faculty of Science of the University of Helsinki and the Academy of Finland grant 322795. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101020762).

REFERENCES

[1] A. Durand, M. Hannula, J. Kontinen, A. Meier, and J. Virtema. Probabilistic team semantics. In F. Ferrarotti and S. Woltran, editors, *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings*, volume 10833 of *Lecture Notes in Computer Science*, pages 186–206. Springer, 2018.

[2] A. Durand and J. Kontinen. Hierarchies in dependence logic. *ACM Transactions on Computational Logic (TOCL)*, 13(4):31, 2012.

[3] P. Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.

[4] P. Galliani, M. Hannula, and J. Kontinen. Hierarchies in independence logic. In S. R. D. Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 263–280, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[5] P. Galliani and L. Hella. Inclusion Logic and Fixed Point Logic. In S. R. D. Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 281–295, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, 1979.

[7] D. Geiger, T. Verma, and J. Pearl. Identifying independence in bayesian networks. *Networks*, 20(5):507–534, 1990.

[8] E. Grädel. Capturing complexity classes by fragments of second-order logic. *Theor. Comput. Sci.*, 101(1):35–57, 1992.

[9] E. Grädel. Model-checking games for logics of imperfect information. *Theor. Comput. Sci.*, 493:2–14, 2013.

[10] E. Grädel and J. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.

[11] M. Hannula and L. Hella. Complexity thresholds in inclusion logic. In R. Iemhoff, M. Moortgat, and R. J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 26th International Workshop, WoLLIC 2019, Utrecht, The Netherlands, July 2-5, 2019, Proceedings*, volume 11541 of *Lecture Notes in Computer Science*, pages 301–322. Springer, 2019.

[12] M. Hannula, Å. Hirvonen, J. Kontinen, V. Kulikov, and J. Virtema. Facets of distribution identities in probabilistic team semantics. In F. Calimeri, N. Leone, and M. Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 304–320. Springer, 2019.

[13] M. Hannula and J. Kontinen. Hierarchies in independence and inclusion logic with strict semantics. *J. Log. Comput.*, 25(3):879–897, 2015.

[14] M. Hannula and J. Kontinen. A finite axiomatization of conditional independence and inclusion dependencies. *Inf. Comput.*, 249:121–137, 2016.

[15] M. Hannula, J. Kontinen, J. Virtema, and H. Vollmer. Complexity of propositional logics in team semantic. *ACM Trans. Comput. Log.*, 19(1):2:1–2:14, 2018.

[16] T. Hyttinen, G. Paolini, and J. Väänänen. Quantum team logic and Bell's inequalities. *Rev. Symb. Log.*, 8(4):722–742, 2015.

[17] N. Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999.

[18] J. Kontinen. Coherence and computational complexity of quantifier-free dependence logic formulas. *Studia Logica*, 101(2):267–291, 2013.

[19] J. Kontinen, A. Kuusisto, P. Lohmann, and J. Virtema. Complexity of two-variable dependence logic and if-logic. *Inf. Comput.*, 239:237–253, 2014.

[20] J. Kontinen, A. Kuusisto, and J. Virtema. Decidability of predicate logics with team semantics. In P. Faliszewski, A. Muscholl, and R. Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, volume 58 of *LIPIcs*, pages 60:1–60:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[21] M. Lück. Canonical models and the complexity of modal team logic. *Logical Methods in Computer Science*, 15(2), 2019.

[22] E. Pacuit and F. Yang. Dependence and independence in social choice: Arrow's theorem. In S. Abramsky, J. Kontinen, J. Väänänen, and H. Vollmer, editors, *Dependence Logic, Theory and Applications*, pages 235–260. Springer, 2016.

[23] R. Rönnholm. Capturing k-ary existential second order logic with k-ary inclusion-exclusion logic. *Ann. Pure Appl. Log.*, 169(3):177–215, 2018.

[24] R. Rönnholm. The expressive power of k-ary exclusion logic. *Ann. Pure Appl. Log.*, 170(9):1070–1099, 2019.

[25] J. Väänänen. *Dependence Logic*. Cambridge University Press, 2007.